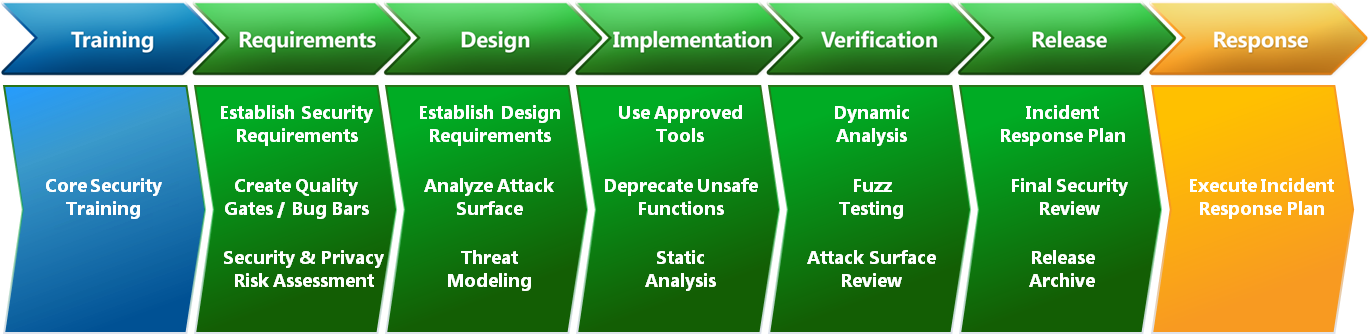
MS Security Development Lifecycle (SDL)

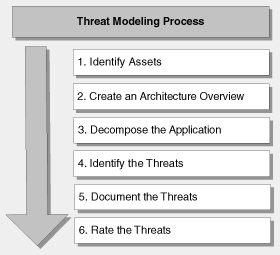
[Copyright © Microsoft Corporation. All rights reserved.](https://msdn.microsoft.com/en-US/library/Ee796683)



# Chapter 3 – Threat Modeling

* [Improving Web Application Security: Threats and Countermeasures - June 2003](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10))

*Figure 1. Secure software development process model at Microsoft*



**The STRIDE Threat Model**

When you are considering threats, it is useful to ask questions such as these:

* How can an attacker change the authentication data?
* What is the impact if an attacker can read the user profile data?
* What happens if access is denied to the user profile database?

You can group threats into categories to help you formulate these kinds of pointed questions. One model you may find useful is STRIDE, derived from an acronym for the following six threat categories:

* **Spoofing identity**. An example of identity spoofing is illegally accessing and then using another user's authentication information, such as username and password.
* **Tampering with data**. Data tampering involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet.
* **Repudiation**. Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise—for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. **Nonrepudiation** refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package.
* **Information disclosure**. Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers.
* **Denial of service**. Denial of service (DoS) attacks deny service to valid users—for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability.
* **Elevation of privilege**. In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed.

### DREAD

The problem with a simplistic rating system is that team members usually will not agree on ratings. To help solve this, add new dimensions that help determine what the impact of a security threat really means. At Microsoft, the DREAD model is used to help calculate risk. By using the DREAD model, you arrive at the risk rating for a given threat by asking the following questions:

* **D**amage potential: How great is the damage if the vulnerability is exploited?
* **R**eproducibility: How easy is it to reproduce the attack?
* **E**xploitability: How easy is it to launch an attack?
* **A**ffected users: As a rough percentage, how many users are affected?
* **D**iscoverability: How easy is it to find the vulnerability?

**3.6 Thread Rating Table**

|  | **Rating** | **High (3)** | **Medium (2)** | **Low (1)** |
| --- | --- | --- | --- | --- |
| D | Damage potential | The attacker can subvert the security system; get full trust authorization; run as administrator; upload content. | Leaking sensitive information | Leaking trivial information |
| R | Reproducibility | The attack can be reproduced every time and does not require a timing window. | The attack can be reproduced, but only with a timing window and a particular race situation. | The attack is very difficult to reproduce, even with knowledge of the security hole. |
| E | Exploitability | A novice programmer could make the attack in a short time. | A skilled programmer could make the attack, then repeat the steps. | The attack requires an extremely skilled person and in-depth knowledge every time to exploit. |
| A | Affected users | All users, default configuration, key customers | Some users, non-default configuration | Very small percentage of users, obscure feature; affects anonymous users |
| D | Discoverability | Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable. | The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use. | The bug is obscure, and it is unlikely that users will work out damage potential. |